Luke Mueller
lam908@mail.harvard.edu
CS181-S16

Assignment #1

Due: 5:00pm February 5, 2016

Collaborators: Nicole Lee, Dong Yan

Homework 1: Linear Regression

You should submit your answers as a PDF via the Canvas course website. There is a mathematical component and a programming component to this homework.You may collaborate with others, but are expected to list collaborators, and write up your problem sets individually.

Please type your solutions after the corresponding problems using this LaTeX template, and start each problem on a new page.

Problem 1 (Centering and Ridge Regression, 7pts)
Consider a data set in which each data input vector $\boldsymbol{x} \in \mathbb{R}^m$. Let $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ be the input matrix, the rows of which are the input vectors, and the columns of which are centered at 0. Let λ be a positive constant. We define:

$$J(\boldsymbol{w}, w_0) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} - w_0\mathbf{1})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} - w_0\mathbf{1}) + \lambda \boldsymbol{w}^T\boldsymbol{w}$$

(a) Compute the gradient of $J(\boldsymbol{w}, w_0)$ with respect to w_0. Simplify as much as you can for full credit.

(b) Compute the gradient of $J(\boldsymbol{w}, w_0)$ with respect to $\boldsymbol{w}$. Simplify as much as you can for full credit. Make sure to give your answer in matrix form.

(c) Suppose that $\lambda > 0$. Knowing that J is a convex function of its arguments, conclude that a global optimizer of $J(\boldsymbol{w}, w_0)$ is

$$w_0 = \frac{1}{n}\sum_i y_i \tag{1}$$

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{2}$$

Before taking the inverse of a matrix, prove that it is invertible.

Solution

$J(\boldsymbol{w}, w_0) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} - w_0\mathbf{1})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} - w_0\mathbf{1}) + \lambda \boldsymbol{w}^T\boldsymbol{w}$

$$= \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}^T w_0\mathbf{1} - (\boldsymbol{X}\boldsymbol{w})^{\mathbf{T}}\boldsymbol{y} + (\boldsymbol{X}\boldsymbol{w})^{\mathbf{T}}(\boldsymbol{X}\boldsymbol{w}) + (\boldsymbol{X}\boldsymbol{w})^{\mathbf{T}}\mathbf{w_0 1} - \mathbf{w_0 1}\boldsymbol{y}^{\mathbf{T}} + \mathbf{w_0 1^T}(\boldsymbol{X}\boldsymbol{w}) + \mathbf{w_0^2 1^T 1} + \breve{}\boldsymbol{w}^{\mathbf{T}}\boldsymbol{w}$$

$\nabla J_{w_0} = -\boldsymbol{y}^T\mathbf{1} + (\boldsymbol{X}\boldsymbol{w})^{\mathbf{T}}\mathbf{1} - \mathbf{1^T}\boldsymbol{y} + \mathbf{1^T}(\boldsymbol{X}\boldsymbol{w}) + \mathbf{2w_0 1^T 1}$

Then, $\boldsymbol{X}^T\mathbf{1} = \mathbf{1^T}\boldsymbol{X} = \mathbf{0}$ since we are taking the sum of columns, which are centered at 0. This also implies $(\boldsymbol{X}\boldsymbol{w})^T\mathbf{1} = \boldsymbol{w}^{\mathbf{T}}\mathbf{X^T 1} = \mathbf{0}$. In addition, $-\boldsymbol{y}^T\mathbf{1} = -\mathbf{1^T}\boldsymbol{y} = -\sum_\mathbf{i} \mathbf{y_i}$, and $\mathbf{1^T 1} = \mathbf{n}$ thus we have:

$$\nabla J_{w_0} = -2\sum_i y_i + 2w_0 n$$

For part (b), first we extract terms relevant to $\nabla J_{\boldsymbol{w}}$:

$-\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} - (\boldsymbol{X}\boldsymbol{w})^T\boldsymbol{y} + (\boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{X}\boldsymbol{w}) + (\boldsymbol{X}\boldsymbol{w})^T w_0\mathbf{1} + \mathbf{w_0 1^T}(\boldsymbol{X}\boldsymbol{w}) + \breve{}\boldsymbol{w}^{\mathbf{T}}\boldsymbol{w}$

Then as before, and since w_0 is a scalar, we have $(\boldsymbol{X}\boldsymbol{w})^T w_0\mathbf{1} = \mathbf{w_0 1^T}(\boldsymbol{X}\boldsymbol{w}) = \mathbf{0}$. Taking the remaining partial derivatives and combining terms the gradient is thus:

$$\nabla J_{\boldsymbol{w}} = -2\boldsymbol{y}^T\boldsymbol{X} + 2\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X} + 2\lambda\boldsymbol{w}^T$$

For part (c) we first return to the gradient of ∇J_{w_0}. Since we have the partial derivative of J with respect to w_0, we can determine the global optimizer of $J(\boldsymbol{w}, w_0)$ by setting the gradient equal to 0. Thus we have:

$$-2\sum_i y_i + 2w_0 n = 0 \Rightarrow -2\sum_i y_i = 2w_0 n \Rightarrow \sum_i y_i = w_0 n \Rightarrow w_0 = \frac{1}{n}\sum_i y_i$$

Solving for $\boldsymbol{w}$ is a similar process, though slightly less trivial. First, we prove $\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}$ is invertible for a later purpose, using the following property: if $det(A) \neq 0$, then A is invertible.

$$det(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}) = det(\boldsymbol{X}^T\boldsymbol{X}) + det(\lambda\boldsymbol{I}) = det(\boldsymbol{X}^T)det(\boldsymbol{X}) + det(\lambda\boldsymbol{I})$$

Then, since we previously showed that $\boldsymbol{X}^T\mathbf{1} = \mathbf{0}$ it follows that there exists a nontrivial solution to $det(\boldsymbol{X})$, thus $det(\boldsymbol{X}) = 0$. Therefore:

$$det(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}) = 0 + det(\lambda\boldsymbol{I}) = \lambda \neq 0$$

$\Rightarrow \boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}$ is invertible.

Now returning to $\nabla J_{\boldsymbol{w}}$, we set the gradient equal to 0 to determine the global optimizer.

$$\nabla J_{\boldsymbol{w}} = -2\boldsymbol{y}^T\boldsymbol{X} + 2\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X} + 2\lambda\boldsymbol{w}^T = 0$$

$$-\boldsymbol{y}^T\boldsymbol{X} + \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{w}^T = 0$$

$$\boldsymbol{y}^T\boldsymbol{X} = \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{w}^T$$

$$\boldsymbol{X}^T\boldsymbol{y} = \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} + \lambda\boldsymbol{w}$$

$$\boldsymbol{X}^T\boldsymbol{y} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\mathbf{I})\boldsymbol{w}$$

$$(\boldsymbol{X}^T\boldsymbol{X} + \lambda\mathbf{I})^{\mathbf{-1}}\mathbf{X^T}\boldsymbol{y} = \boldsymbol{w}$$

Problem 2 (Priors and Regularization,7pts)
Consider the Bayesian linear regression model given in Bishop 3.3.1. The prior is

$$p(\boldsymbol{w} \mid \alpha) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{0}, \alpha^{-1}\boldsymbol{I}),$$

where α is the precision parameter that controls the variance of the Gaussian prior. The likelihood can be written as

$$p(\boldsymbol{t} \mid \boldsymbol{w}) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid \boldsymbol{w}^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1}),$$

Using the fact that the posterior is the product of the prior and the likelihood (up to a normalization constant), show that maximizing the log posterior (i.e., $\ln p(\boldsymbol{w} \mid \boldsymbol{t}) = \ln p(\boldsymbol{w}|\alpha) + \ln p(\boldsymbol{t} \mid \boldsymbol{w})$) is equivalent to minimizing the regularized error term given by $E_D(\boldsymbol{w}) + \lambda E_W(\boldsymbol{w})$ with

$$\begin{aligned} E_D(\boldsymbol{w}) &= \frac{1}{2} \sum_{n=1}^{N} (t_n - \boldsymbol{w}^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{x}_n))^2 \\ E_W(\boldsymbol{w}) &= \frac{1}{2} \boldsymbol{w}^{\mathsf{T}} \boldsymbol{w} \end{aligned}$$

Do this by writing $\ln p(\boldsymbol{w} \mid \boldsymbol{t})$ as a function of $E_D(\boldsymbol{w})$ and $E_W(\boldsymbol{w})$, dropping constant terms if necessary. Conclude that maximizing this posterior is equivalent to minimizing the regularized error term given by $E_D(\boldsymbol{w}) + \lambda E_W(\boldsymbol{w})$. (Hint: take $\lambda = \alpha/\beta$)

Solution

Starting with the prior:

$$p(\boldsymbol{w} \mid \alpha) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{0}, \alpha^{-1}\boldsymbol{I}) = \frac{1}{\sqrt{2\pi\alpha^{-1}\boldsymbol{I}}} e^{-(\boldsymbol{w}-\boldsymbol{0})^2 \big/ 2\alpha^{-1}\boldsymbol{I}}$$

Taking the natural log and removing irrelevant terms (i.e. constants, since the posterior need only be proportional) we are left with:

$$\frac{(\boldsymbol{w}-\boldsymbol{0})^{\boldsymbol{2}}}{2\alpha^{-1}\boldsymbol{I}} = \frac{1}{2\alpha^{-1}\boldsymbol{I}} \boldsymbol{w}^{\mathsf{T}} \boldsymbol{w} = \frac{E_W(\boldsymbol{w})}{\alpha^{-1}\boldsymbol{I}}$$

Now for the likelihood:

$$p(\boldsymbol{t} \mid \boldsymbol{w}) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid \boldsymbol{w}^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1}) = \frac{1}{\sqrt{2\pi\beta^{-1}}} e^{-\left(t_n - \boldsymbol{w}^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{x}_n)\right)^2 \big/ 2\beta^{-1}}$$

As before, taking the natural log and removing irrelevant terms we are left with:

$$\frac{1}{2\beta^{-1}} \sum_{n=1}^{N} (t_n - \boldsymbol{w}^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{x}_n))^2 = \frac{E_D(\boldsymbol{w})}{\beta^{-1}}$$

Now combining the prior and likelihood through the sum of natural logs we have:

$$\ln p(\boldsymbol{w} \mid \boldsymbol{t}) = \frac{E_D(\boldsymbol{w})}{\beta^{-1}} + \frac{E_W(\boldsymbol{w})}{\alpha^{-1}\boldsymbol{I}}$$

Then multiplying by β^{-1} and defining $\lambda = \alpha/\beta$ we have:

$$\ln p(\boldsymbol{w} \mid \boldsymbol{t}) = E_D(\boldsymbol{w}) + \lambda E_W(\boldsymbol{w})$$

3. Modeling Changes in Congress [10pts]

The objective of this problem is to learn about linear regression with basis functions by modeling the average age of the US Congress. The file `congress-ages.csv` contains the data you will use for this problem. It has two columns. The first one is an integer that indicates the Congress number. Currently, the 114th Congress is in session. The second is the average age of that members of that Congress. The data file looks like this:

```
congress,average_age
80,52.4959
81,52.6415
82,53.2328
83,53.1657
84,53.4142
85,54.1689
86,53.1581
87,53.5886
```

and you can see a plot of the data in Figure 1.

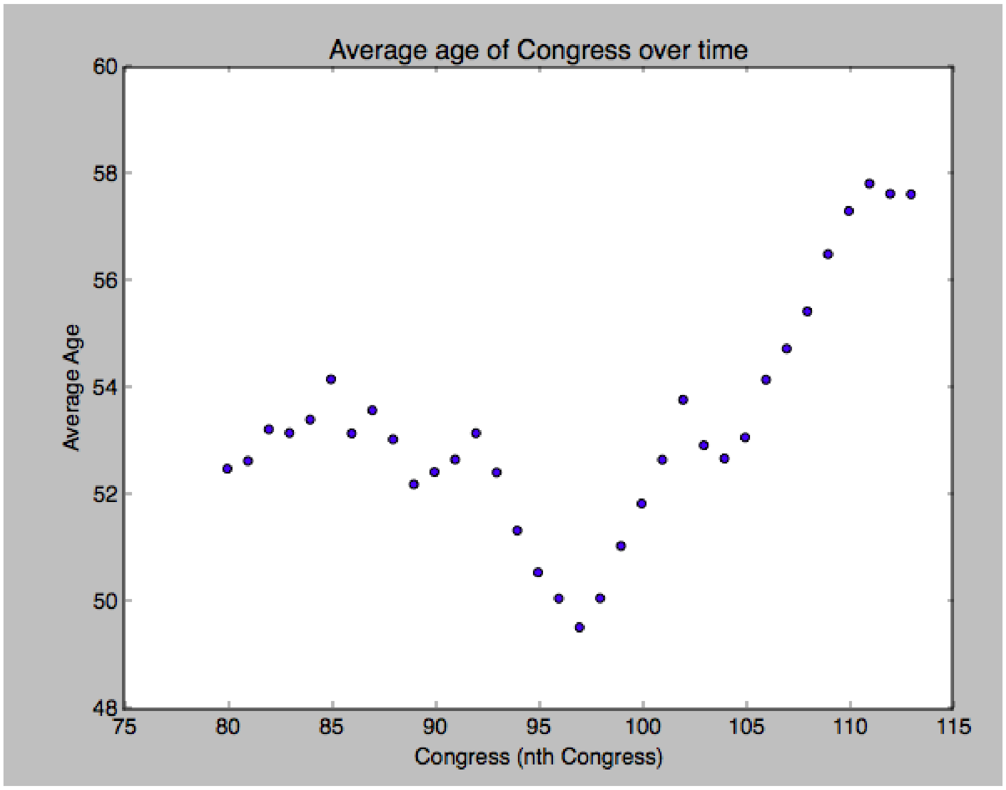

Figure 1: Average age of Congress. The horizontal axis is the Congress number, and the vertical axis is the average age of the congressmen.

Problem 3 (Modeling Changes in Congress, 10pts)

Implement basis function regression with ordinary least squares with the above data. Some sample Python code is provided in `linreg.py`, which implements linear regression. Plot the data and regression lines for the simple linear case, and for each of the following sets of basis functions:

(a) $\phi_j(x) = x^j$ for $j = 1, \ldots, 7$

(b) $\phi_j(x) = x^j$ for $j = 1, \ldots, 3$

(c) $\phi_j(x) = \sin\{x/j\}$ for $j = 1, \ldots, 4$

(d) $\phi_j(x) = \sin\{x/j\}$ for $j = 1, \ldots, 7$

(e) $\phi_j(x) = \sin\{x/j\}$ for $j = 1, \ldots, 20$

In addition to the plots, provide one or two sentences for each, explaining whether you think it is fitting well, overfitting or underfitting. If it does not fit well, provide a sentence explaining why. A good fit should capture the most important trends in the data.

Solution

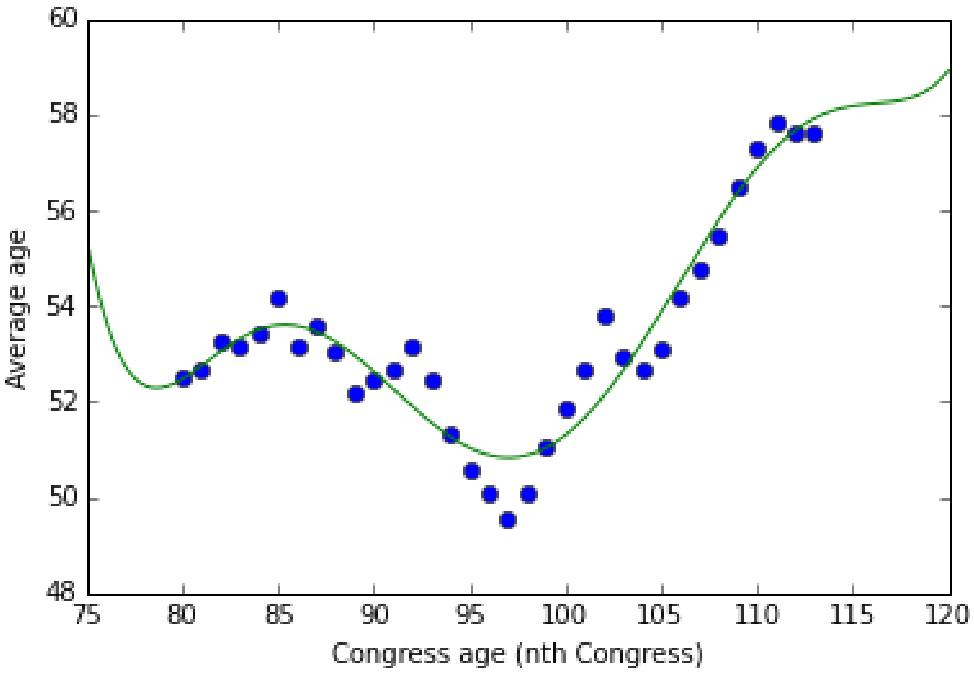

Figure 2: Graph for part A. The fit appears to do well in this plot, except in the lower tail where the data do not reflect the trend line. However, the main concern here is overfitting. We have only 34 data points, yet we are using 7 coefficients to perform our regression.

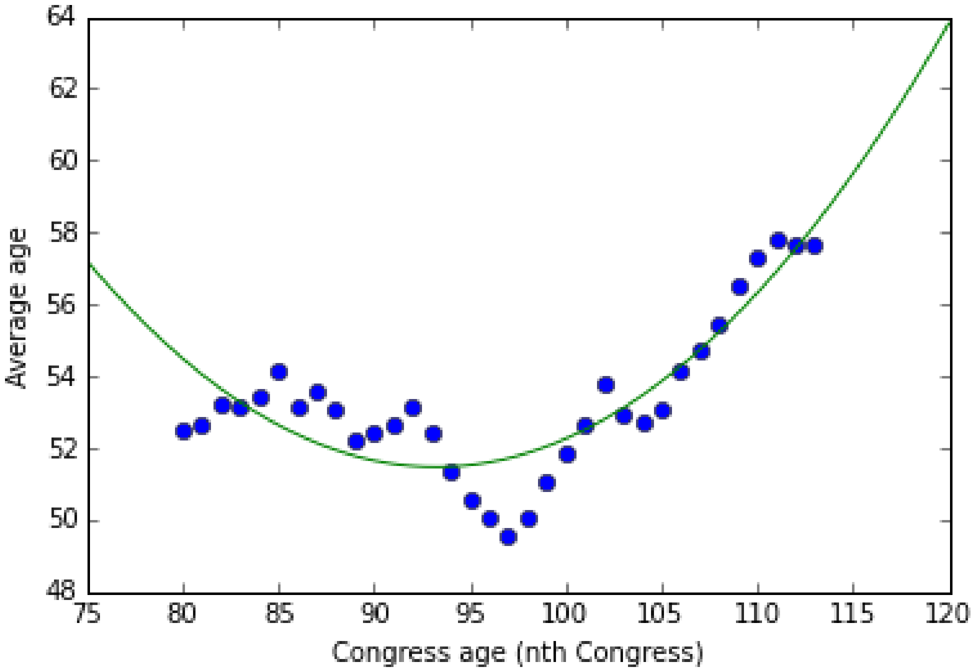

Figure 3: Graph for part B. The fit is not as good as in part (a), and we definitely don't seem to be capturing any useful information in the tails (which seem to plateau somewhat). But here we use far fewer degrees of freedom (coefficients) to fit the regression, which is a plus.

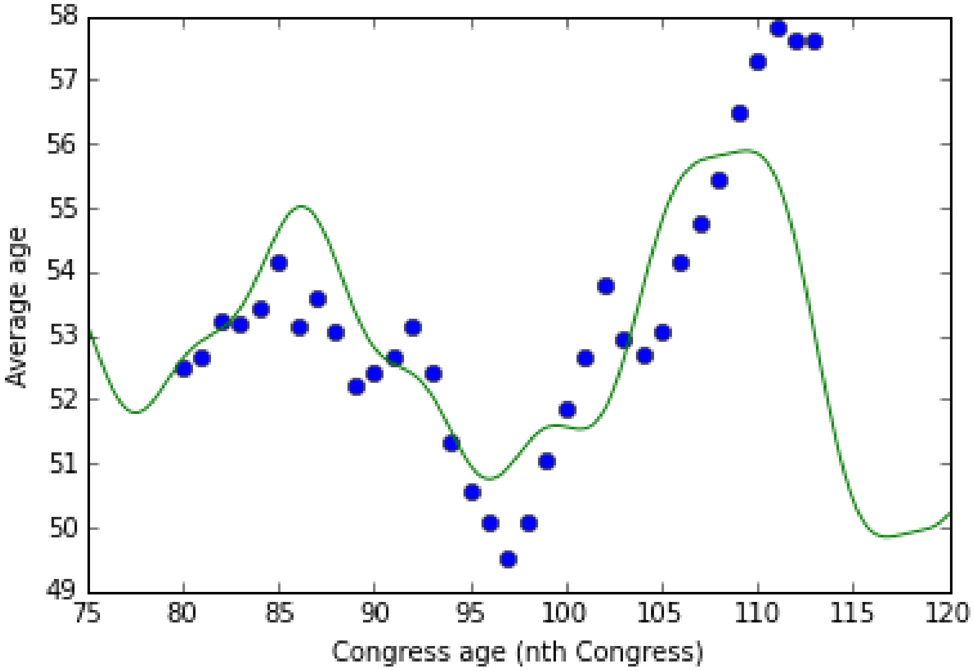

Figure 4: Graph for part E. This plot is more or less a compromise of the previous plots, since it uses fewer degrees of freedom than (a) yet provides slightly better fit than (b). However, it seems like we could get by with fewer degrees of freedom since there are some unnecessary ”wiggles” in the regression line. Also, there is a disconnect between the data and the regression line in the tails.

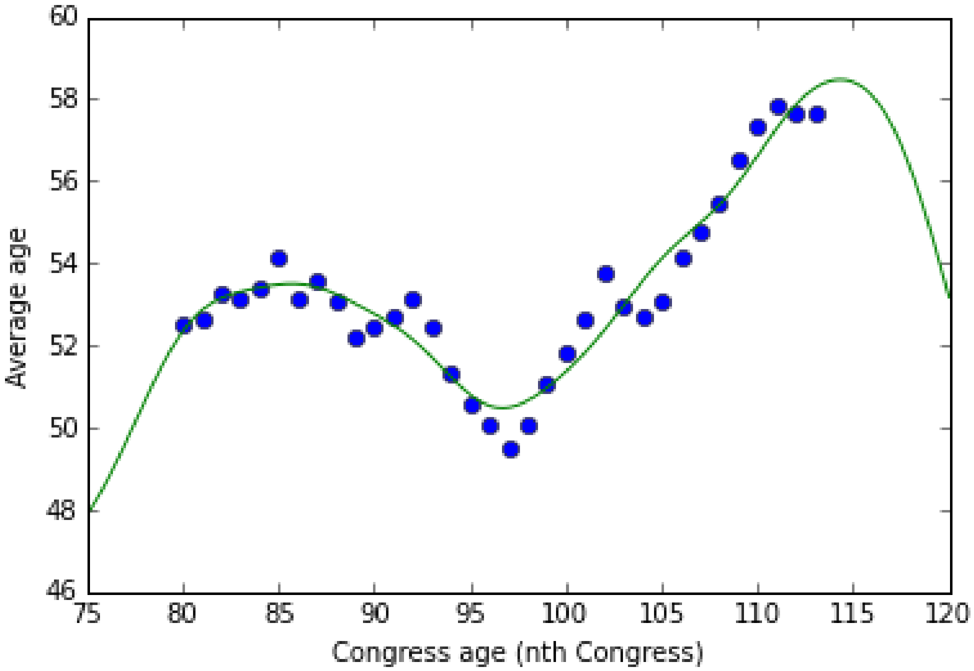

Figure 5: Graph for part D. This plot is more accurate than plot (c), capturing the information in the tails far better than before. However, as in part (a) we are using many degrees of freedom so there is concern for overfitting.

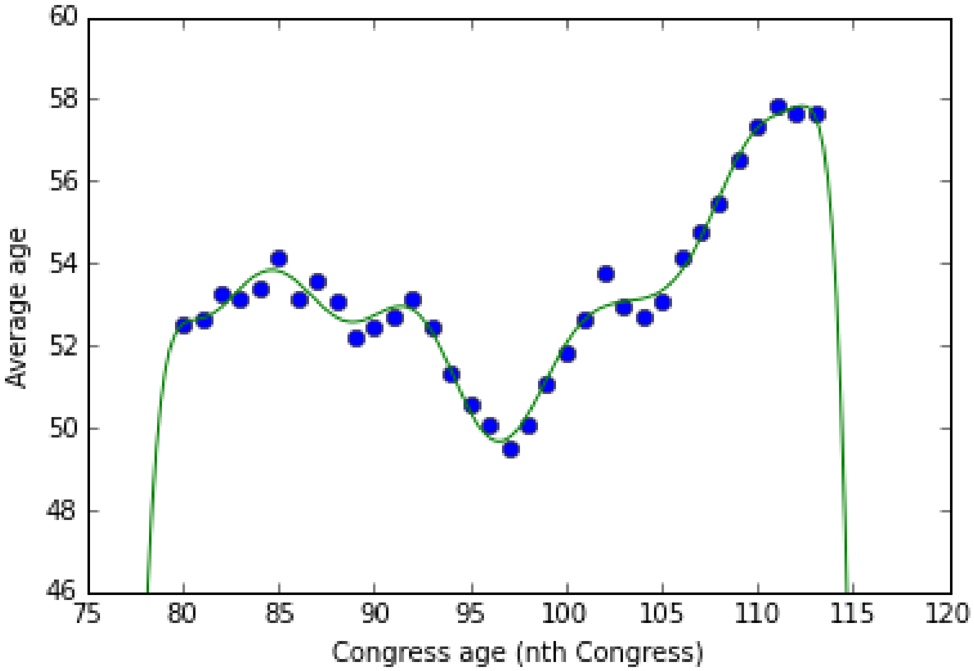

Figure 6: Graph for part E. Obviously, this is the best fit of any plot so far. With so many degrees of freedom, we capture nearly every point perfectly. However, the overfitting is too much to make any meaningful inference. Perhaps the only use here is identifying potential outliers (i.e., between 100-105th congress).

Problem 4 (Calibration, 1pt)

Approximately how long did this homework take you to complete?

Answer:
7 hours. 2-3 hours LaTeX (beginner), 1-2 hours python, 3-4 (or more) working out problem 1 and 2. My linear algebra is very rusty!